GCCS/DII COE System Integration Support

DII COE Segment System Administrator's Manual (for the SeComp Tool Version 1.0.0.3)

May 21, 1997

Prepared for:

DISA/JEJA
ATTN: Claire Burchell
45335 Vintage Park Plaza
Sterling, VA 20166-6701

Contract Number: DCA100-94-D-0014
Delivery Order Number: 330, Task 6
CDRL Number: A005

Prepared by:

Computer Sciences Corporation
Defense Enterprise Integration Services
Four Skyline Place
5113 Leesburg Pike, Suite 700
Falls Church, VA 22041

THIS DOCUMENT IS UNCLASSIFIED

Defense Information Infrastructure (DII)

Common Operating Environment (COE)

Segment System Administrator's Manual for the Security Compliance (SeComp) Tool Version 1.0.0.3

May 21, 1997

Prepared for:

DISA/JEJA
ATTN: Claire Burchell
45335 Vintage Park Plaza
Sterling, VA 20166-6701

Prepared by:

Computer Sciences Corporation
Defense Enterprise Integration Services
Four Skyline Place
5113 Leesburg Pike, Suite 700
Falls Church, VA 22041

Table of Contents

Preface

The following conventions are used in this document:

Bold	Used for information that is typed, pressed, or selected in executables and instructions. For example, select **connect to host**.
Italics	Used for file names, directories, scripts, commands, user IDs, document names, and Bibliography references; and any unusual computerese the first time it is used in text.
Underline	Used for emphasis.
Arrows <>	Used to identify keys on the keyboard. For example <Return>.
"Quotation Marks"	Used to identify informal, computer-generated queries and reports, or coined names; and to clarify a term when it appears for the first time. For example "Data-Generation Report."
`Courier Font`	Used to denote anything as it appears on the screen or command lines. For example `tar xvf dev/rmt/3mm`.
Capitalization	Used to identify keys, screen icons, screen buttons, field, and menu names.

1. System Administrator's Manual (SAM) for the SeComp Tool

The Defense Information Infrastructure (DII) Common Operating Environment (COE) Security Compliance (SeComp) tool provides functions and procedures to determine if, where, when, and how COE and application segments[1] might be disruptive to a system security configuration. The recommended (advisory) security configuration for the DII COE is based on the DII COE Security Checklists. This document will describe SeComp use considering the DII COE and the DII Security Checklist. The SeComp tool examines and reports on the configuration items identified in the DII Security Checklists. The SeComp tool will continue to be updated as the DII Security Checklist is updated or additional security-relevant checks are required.

The SeComp tool is currently capable of executing on both the Solaris and HP-UX platforms. There are two versions of the software, one for Solaris 2.5 and one for HP 10.x. Additionally, the SeComp tool will be ported and integrated for upcoming versions of other UNIX-based platforms certified for DII COE operation as required. The SeComp tool will not be used for the NT platform.

The SeComp tool is not segmented. The platform-dependent versions are provided via *tar* format file component within the Developers Toolkit. The reasoning for this is to allow use in an non-segmented laboratory environment[2]. This document will provide overview, execution and analysis, configuration and installation instructions, and guidance.

2. SeComp Procedural Overview

The purpose of the SeComp tool is four-fold:

- The SeComp tool will be used to prepare a COE suite to ensure that the COE suite is in the DII recommended security configuration at the beginning of a segment security compliance test operation.

- The SeComp tool will be used to ensure that the COE suite remains in the recommended security configuration at the end of a successful[3] segment installation

1 System mission application segments are beyond the scope of the DII COE for security compliance checking since each mission is controlled by separate security policy and requirements. However, product segment developers and providers and system Information Systems Security Officer's (ISSO's) will have access to the SeComp software via the DII COE Developers Toolkit. Security compliance checking should be incorporated with the segment verifications that currently take place.

2 The SeComp tool is used strictly in the confines of either the development or integration laboratory.

3 It is not reasonable to security test segments that fail functional (as opposed to security) element testing.

- The SeComp tool will be used to ensure that the COE remains in the recommended security configuration at the end of a successful segment test execution.

- The SeComp tool will be used to ensure that the COE remains in the recommended security configuration at the end of a successful segment de-installation process.

The SeComp tool is accompanied by this set of SeComp guidelines. The SeComp tool provides a set of reports that are analyzed to detect modifications that a COE kernel or segment may have imposed on the recommended DII COE security configuration.

2.1 SeComp Testing and Analysis

The SeComp tool is designed to be executed against an isolated segment[4] residing on a file system tuned to a controlled and known security configuration. The tool may be used during two processes that are currently defined for the verification of segments delivered to the DII COE Configuration Management (CM); the integration testing and quality assurance (QA) testing processes. The primary difference in the two processes is that the integration process primarily checks COE integration compatibility issues, whereas the QA process primarily checks runtime application operation.

The next two sub-sections provide a procedural example as to how the testing and reporting results might occur. However testing is accomplished, it is important that the findings for each segment be completely and accurately documented.

2.1.1 Integration Test Process

The integration process is where SeComp testing begins. The integration process focus will be to identify integration and compatibility issues up to the point of full operational execution through these four steps:

Step 1: Prepare and configure the test platform.

- Execute SeComp with the *-b* (initial) option, setup the test platform(s) in accordance with the DII Security Checklist.

- Remove master files[5], re-run SeComp with the *-b* (initial) option, collect reports.

4 Isolation must be identified to what is reasonable given a segment's prerequisite requirements.

5 Removing the master files will cause the them to be re-generated, thus representing the true configuration after required modifications.

Step 2: Install the segment.

- Run SeComp with the *-I* (install) option, collect reports.

Step 3: Execute and immediately close the segment.

- Run SeComp with the *-e* (execute) option, collect reports.

Step 4: De-install the segment.

- Run SeComp with the *-z* (de-install) option, collect reports.

The integration analysts will determine if the recommended security configuration has been affected, identify the specific discrepancies, and make recommendations concerning the importance of the discrepancies. The segment, test reports, and test comments are then passed to the DII Integration Security Officer to perform additional analysis and obtain clarifications (if required) from the integration analysts. The DII Integration Security Officer will then make recommendations and pass the package to the appropriate DII Designated Approval Authority (DAA) for final analysis, risk determination, and segment disposition.

2.1.2 Quality Assurance Test Process

The QA process performs its testing using the most recent versions of the DII segments and therefore provides the most robust testing environment suitable for application operation testing. The QA analysts will execute the SeComp tool in a similar manner described for the integration process, with reduced emphasis on Steps 2 and 4. and increased emphasis on Step 3 where a complete runtime execution test is completed for the segment:

Step 1: Prepare the test platform.

- Execute SeComp with the *-b* (initial) option, setup the test platform(s) in accordance with the DII Security Checklist.

- Remove master files, re-run SeComp with the *-b* (initial) option, collect reports.

Step 2: Install the segment.

Step 3: Execute and completely test the application's operational functions.

- Run SeComp with the *-e* (execute) option, collect reports.

Step 4: De-install the segment[6].

The QA analyst will determine if the recommended security configuration has been affected, identify the specific discrepancies, and make recommendations concerning the importance of the discrepancies. The segment, test reports, and test comments are then passed to the DII Security Engineer to perform additional analysis and obtain clarifications (if required) from the integration analysts. The DII Security Engineering Officer will then make recommendations and pass the package to the appropriate DII DAA for final analysis, risk determination, and segment disposition.

There are a number of modifications that might be made to the processes just described, for example, combining the integration and QA reporting processes for expediency. It is important that the process is smoothly integrated and does not become a bottleneck in the overall segment verification and testing process.

2.2 Segment Risk Analysis and Disposition

The integration and QA analysts forward the SeComp report package to the DII Integration Security Officer and the DII Engineering Security Officer for risk analysis. This risk analysis team will examine the findings and assess them regarding any discrepancies that may have been found. It is expected that most segments will comply with the security configuration, and in these cases, the risk analysis team will only need to confirm the findings.

It is recommended that segments should be approved for distribution only after the risk analysis team has indicated that the new segment been accepted and has included any statements regarding the conditions pertaining to the acceptance.

It is recommended that guidelines be established that bound the limits of what may be determined acceptable. For example, object permissions of security-related features at the operating system (OS) level should be considered untouchable. Further, objects that have been known to usurp root privilege if adequate protections are not maintained should be considered in a like manner.

It is recommended that the risk analysis team be delegated an appropriate level of authority to perform its mission. The level of authority should be sufficient to reject a segment on the basis that it would incur unreasonable risk to the security posture of the DII. The basis for these decisions should be documented and be required to reference the appropriate security policy and requirements statements. The documentation identifying the cause for rejection and the segment can then be returned to the segment developer/provider.

6 De-installing may not be required during the QA phase.

3. SeComp Test Procedure

This section will amplify the steps outlined in Section 2.1. This section will treat both the integration and QA process testing generically and focus on describing the details of each step in the four step process.

The *report* and *master* files (described in Section 4.5) are created when SeComp is executed. The *report* and *master* files must be manually analyzed to determine differences between the reports.

3.1 Establishing the DII Recommended Security Configuration

The first step in the security compliance process is to establish the DII COE recommended security configuration on the target COE test host platform. It is recommended that only the segments and software required for the testing of the segment be installed on the test platform to isolate the segment to be tested as much as possible.

3.1.1 The DII COE Security Configuration

The DII COE recommended security configuration is based on the DII Security Checklist. Once the recommended configuration is established, the SeComp reports are used to identify areas that are out-of-compliance with the DII Security Checklist.

The DII COE security configuration is based on a common-sense approach to security configuration. The underlying philosophy of the DII COE computer security configuration will focus on the following strategies listed in priority of importance:

- Protecting and controlling root privilege is paramount.
- Controlling access to the interconnected system components begins with access point identity and applying the appropriate controls. Access control is maintained through pro-active and re-active configuration and audit trail monitoring.
- File system integrity is ultimately and more effectively gained through sensible discretionary access control (DAC) and user privilege control (through user account and profiling control mechanisms) management.
- Only the file system objects and executable processes that are required to complete the operational mission are maintained on the file system.
- Primary misuse detection is accomplished through the maintenance of strategies 1 through 4[7].

7 This is not meant to say that additional specialized detection mechanisms should not be implemented. Higher security environments may even require these mechanisms. For many environments, however, pro-active monitoring of the system configuration and audit logs will provide the information required to detect misuse attempts and assess damages by

- Establish a standard countermeasure plan or operating procedure to execute when misuse is detected. Once abnormal behavior is detected, the countermeasure procedure is followed that identifies the who, what, when, where, and why of the access, corrects the anomalies allowing the access, and then assesses the damages.

The DII COE is an open-system concept and infrastructure to the point that the information and resources that are necessary for mission completion are as transparently and seamlessly available as possible. The DII COE security philosophy does not conflict with the open system concept and is maintained in a manner that supports both infrastructure security and system operational capabilities. Additionally, the DII COE security philosophy does not conflict with critical period operations (emergency or life-and-death operational scenarios) and is capable of giving way to mission considerations during these times. While giving way to these special considerations and system requirements, however, the DII COE security philosophy and resulting security configuration must be and is capable of being implemented to exhibit a low tolerance for promoting abusive or damaging behaviors, whether performed accidentally or with malice.

The DII COE Security Checklist provides configurations of system elements and features that provide basic security for an interconnected system. The basic security provided may not be enough for the high security requirements of some systems, however, the basic security provided will not impose undue restrictions or limitations on systems with lower or no security requirements. Proper configuration and ongoing maintenance is in all cases required to meet the criteria of low and high security requirements.

3.1.2 Configuring SeComp

When configuring the SeComp features and when examining the SeComp reports, the DII COE security philosophy and strategies combined with the information provided in the DII Security Checklists should be the reference point to determine the validity of a reported item.

The SeComp tool can be configured to examine the system and report specific discretionary access assignments or look for the existence of certain objects on the file system. The configurable features that are available are discussed in Section 4. Once the configurable features are set, the SeComp tool can more accurately provide the information required to analyze the segment's affect on the security configuration being tested. Once the configuration is complete, the SeComp tool is executed.

Once SeComp has executed in the initial phase, the generated reports are analyzed. The SeComp reports and master files will identify areas that are out-of-configuration with the recommended security configuration. Before continuing to install the segment to be tested, the security configuration analysts must first bring the system into the recommended configuration by modifying the out-of-configuration details (permission sets, unauthorized network daemon

examining the historical system records. In fact, this method is sufficient for most system-high classified system implementations when rigorously applied and adapted to a systems security policy.

activation, etc.). Once the out-of-configuration details have been brought into configuration, the security configuration analysts will delete the existing set of master files (see Section 4.5.1) generated during this initial run. Once complete, the SeComp tool is re-executed with the *-b* (initial) option to generate a new set of master files and reports. The reports should be analyzed to confirm that the configuration is now in compliance. Once the initial compliance is ensured, testing may begin for a COE segment. The set of reports and master files just generated reflect the desired state of the security configuration and are referred to as the security baseline set of reports.

3.2 Install and Test the Target COE Segment

The second step is to install the COE segment to be tested. It is imperative to follow the installation instructions verbatim during this process in order to accurately identify problem areas. Immediately following the complete and successful installation, the SeComp tool is executed again, this time with the *-I* (install) option. It is imperative that no other operations occur between the segment installation and SeComp execution in order to avoid configuration modifications that might incorrectly be blamed on the segment.

After SeComp has executed, the generated reports are analyzed. This set of reports are referred to as the installation phase reports; no new master files are generated. The SeComp installation reports are analyzed and, by analyzing the installation phase reports and comparing them to the security baseline reports, the analyst will be able to identify the differences and areas that are out-of-configuration with the recommended security configuration. The security configuration analyst must document any out-of-configuration details that are identified in the SeComp report before continuing the analysis of the segment. After all out-of-configuration details have been documented, the analyst may proceed.

3.3 Execute and Test the Target COE Segment

The third step occurs after the COE segment has been functionally tested[8]. It is imperative to follow the operational instructions verbatim during this process in order to identify specific problem areas.

After the COE segment has been tested, execute the SeComp tool again, this time with the *-e* (execute) option, and analyze the reports generated. This set of reports are referred to as the execution phase reports; no new master files are generated. The SeComp execution reports are analyzed and, by analyzing the execution phase reports and comparing them to the security baseline reports, the analyst will be able to identify the differences and areas that are out-of-configuration with the recommended security configuration. It may also be required to refer to the installation phase reports to complete the analysis for the execution phase. The security configuration analyst must document any out-of-configuration details that are identified in the

8 Recall that functional testing in the integration environment is simply executing and closing the segmented application whereas the QA testing is a thorough testing of the segmented application.

SeComp report before completing the analysis of the segment. After all out-of-configuration details have been documented, the analyst may proceed.

3.4 De-Install Target COE Segment

The fourth step begins by de-installing the segment once the COE segment testing has been completed.

After the COE segment has been de-installed, execute the SeComp tool again, this time with the *-z* (de-install) option, and analyze the reports generated. This set of reports are referred to as the de-installation phase reports; no new master files are generated. The SeComp execution reports are analyzed and, by analyzing the de-installation phase reports and comparing them to the security baseline reports, the analyst will be able to identify the differences and areas that are out-of-configuration with the recommended security configuration. It may also be required to refer to the installation and execution phase reports to complete the analysis for the de-installation phase. The security configuration analyst must document any out-of-configuration details that are identified in the SeComp report before completing the analysis of the segment. After all out-of-configuration details have been documented, the analyst may proceed.

3.5 Test Conclusion

After the segment testing has been completed, the security configuration analyst must reset any out-of-configuration parameters that were found up to this point to ensure that the test suite is left in the recommended security configuration. Once the out-of-configuration parameters have been brought into configuration, the COE segment testing may be considered complete, and testing of the next COE segment may begin.

The final step in the SeComp process is to turn over all SeComp reports and accompanying documentation to the risk analysis team.

4. SeComp Implementation Details

The SeComp tool should be installed in a protected hierarchy on the file system, for example, in */etc*. This tool should not be generally accessible to the public to ensure its integrity during operation. It is recommended that the *DII COE SeComp Software Requirements Specification* (SRS) be reviewed in conjunction with this document to enhance knowledge of the SeComp design and operation.

4.1 SeComp Operations

The SeComp tool may be configured and executed on a standalone platform or in a client/server networked setup. The SeComp tool requires root privilege in order to access all of the file system security-relevant configuration details. The SeComp tool is executed via the command line.

The SeComp tool may be executed on any platform in a networked test configuration and its reports may be collected on any platform. The SeComp runtime software and reports do not take up a great deal of memory, however it would be prudent to reserve at least 10 megabytes for its operation.

When running SeComp in a client/server configuration, the server portion is comprised of the SeComp software directories and the platform is configured to "share" the SeComp runtime software directories. The SeComp clients require only to be configured to automount the SeComp software from the platform configured as the SeComp server.

In the client/server configuration, it is possible to test several segments at a time, a segment per client host platform. In this scenario, the processing described in Section 3 is accomplished on each client platform.

The SeComp tool, when properly configured, will produce the following reports:

- Identification and Authentication (I&A) report
- DAC report
- System Configuration (*sys_config*) report
- Audit report
- Password Vulnerability report
- Shared file system report.

In addition to the reports, the SeComp tool creates four master files:

- Configuration Checklist master (*cklist.master.[time stamp]*) file
- UID master (*suid.master.[time stamp]*) file
- World-Writable master report
- Set Unknown UID (file owner and group identities) master report.

Finally, in conjunction with each task report generated, a diff file report is generated. The diff file reports compare the install, execute, and de-install phase reports with the parallel reports generated during the initial phase. Each phase's diff reports are stored in the phase execution directory. The diff reports are named for the task and suffixed with *.diff*, (e.g., for an audit diff report, the file name would be *audit.diff*). The diff reports will be useful in identifying the changes taking place on the file system between SeComp phase executions during a segment security compliance check.

4.2 SeComp Runtime Options

The SeComp program supports eight options:

- *-b* Initial configuration execution.
- *-i* Segment install execution.

C	*-e*	Segment operation execution.
C	*-z*	Segment de-install execution.
C	*-v*	Verbose report output.
C	*-t* “task list”	Tasks to be executed in quotes and space delimited.
C	*-r [path]/reports*	Where the reports directory hierarchy is located. The *reports* directory must exist.
C	*-m [path]/masters*	Where the masters directory hierarchy is located. The *masters* directory must exist.

These options allow flexibility in the configuration of the SeComp program execution and master and report file collection depositories.

4.3 SeComp Commands

There are four commands used with SeComp that reside in the SeComp *runtime* directory (e.g., */etc/secomp*):

C *run_secomp*
C *view_status*
C *view_rep*
C *view_allrep*.

The *run_secomp* script is a shell script program that is used to prepare and execute the SeComp tool. The configuration sections cover the details of this program later, however, to execute, simply enter:

```
cd /etc/secomp
run_secomp [-biezvtrm]
```

The default option (if no options are selected) is the *-b* (initial) option.

The *view_status* is a task status checking script. This program will print the status of the secomp program as found in the *~/secomp/reports/[hostname]/latest/SECOMP_PHASE* report directory. *SECOMP_PHASE* identifies the SeComp execution phase as one of initial, install, execute, or de-install. This program will understand its environment and know where the *reports* directory is. To execute the program from the command line enter:

```
cd /etc/secomp
/view_status
```

The *view_rep* is a script that displays a given report for a given host name. This program will print the report for the named SeComp task as found in the *~/secomp/reports/[hostname]/latest/SECOMP_PHASE* report directory with the suffix *.rpt*. This program will understand its environment and know where the *reports* directory is. This command does have two required arguments, first the report name in the format *task_name.rpt* and, second, *hostname* as it is defined in the system */etc/host* name file for the host whose report is to be reviewed. To execute the program from the command line enter:

```
cd /etc/secomp
/view_rep INA hostname
```

The *view_allrep* is a script that displays all of the reports found for a given host name. This program will print the reports for the SeComp tasks as found in the *~/secomp/reports/[hostname]/latest/SECOMP_PHASE* report directory with the suffix *.rpt*. This program will understand its environment and know where the *reports* directory is. This command does have one required argument, the host name as defined in the system host name tables for the host whose report is to be reviewed. To execute the program from the command line enter:

```
cd /etc/secomp
/view_allrep hostname
```

4.4 SeComp Tasks

The SeComp tool provides the following tasks:

- I&A task
- DAC task
- System Configuration (*sys_config*) task
- Audit (*audit*) task
- Password Vulnerability (I&A) task
- World-Writable (*wwrite*) task
- Unknown UID (*nouser*) task.

4.5 SeComp Report and Master Files Generation

SeComp stores information concerning the configuration details in two hierarchies, the reports and masters directories. The reports are described in the next two sections. A more detailed discussion of the SeComp hierarchy and directory structures may be found in the SRS.

4.5.1 SeComp Report Files

All SeComp reports will be stored in the *~/secomp/reports* directory unless the *reports* directory location is modified in the *runtime.cf* file or by the *-r* command line option. A subdirectory is created by the SeComp processing (after SeComp is executed for the client the first time) for each

client under the *reports* directory and named by the *uname -n* output for the client host. For example, in the default runtime configuration a host named *client1* would find its reports in:

```
/etc/secomp/reports/client1
```

Also, the *reports* directory will contain sub-directories that store the contents of the reports generated. The directory containing the latest reports run will be linked to the directory name *latest*. All of the *report* directories (except the linked directory *latest*) will be named by a time stamp reflecting the execution time.

The time stamp directories are populated with directories that represent each of the SeComp execution phases of initial, install, execute, and de-install. The directories are created at the time of execution of the *run_secomp* program with the given option. The phase directories will contain the reports generated during that phase and will be compared to the initial phase reports for differences.

The actual reports contained in the report sub-directories are named for the task it reports and will be suffixed by *.rpt*, for example, for the *sys_config* task, a report named:

```
/etc/secomp/reports/client1/latest/[SECOMP_PHASE]/sys_config
.rpt
```

would be generated for the *client1* host.

When the password vulnerability check executes as part of the I&A checks, there will be two additional files in the report subdirectory with the somewhat cryptic name of *out.[process_id]*. The *crack* engine names the output files with the process ID of the *crack* process, this will replace the *process_id* portion of the name. For example, assume the process ID of the *crack* process is 8895. The *crack* engine will place the result of the process in a file called:

```
/etc/secomp/reports/client1/latest/[SECOMP_PHASE]/out.8895
```

There may be more than one *.out* file that *crack* generates. The pertinent contents of the *out.[process_id]* files are automatically appended to the I&A report.

4.5.2 SeComp Master Files

All SeComp master will be stored in the *~/secomp/masters* directory unless the master file directory location is modified in the *runtime.cf* file or by the *-m [path]* command line option. A subdirectory is created by the SeComp processing (after SeComp is executed for the client the first time) for each client under the *masters* directory and named by the *uname -n* output for the client host. For example, in the default runtime configuration a host named *client1* would find its master files in:

```
/etc/secomp/masters/client1
```

The actual master files contained in the *masters* sub-directories are named for the master file content and a time stamp that indicates when the master snapshot was taken. Currently, there are two master files created by the SeComp tool, one that lists all security-relevant files configured in the *secomp.cf* file and one that lists all the set UID programs found on the host. These files are named using the following format:

```
/etc/secomp/masters/client1/cklist.master.[time stamp]
/etc/secomp/masters/client1/suid.master.[time stamp]
```

5. SeComp Configuration

The SeComp tool is configured with some details of the desired security configuration. The security analyst must decide and plan:

- The server to be used as the SeComp server.
- The locations of the SeComp depository directories (*reports* and *masters*).
- The SeComp tasks to be executed.
- The NFS file-share setup on the server.
- The *automount* configuration changes required at the clients.

Installation instructions for the SeComp tool may be found in the *DII COE SeComp Installation Procedures* (IP). The next sections outline the configuration details of the SeComp program and identify the default configuration settings.

5.1 SeComp Configuration Decisions

The SeComp configuration consists of three primary decisions defining how the program will be run:

- Where the SeComp runtime software will be located on the server file system.
- Where the SeComp generated report and master file directories will reside.
- The tasks the SeComp tool will execute.

The default configuration is:

- The SeComp runtime software is installed to */etc/secomp*.
- The SeComp generated report hierarchy are in */etc/secomp/reports* and */etc/secomp/masters*.

The SeComp program will keep track of the locations of all the paths it must know concerning its execution by referring to the configuration files that contain this information.

5.2 SeComp Configuration Files

The SeComp configuration files are all located in the SeComp hierarchy and are:

- *~/secomp/runtime.cf*
- *~/secomp/secomp.cf*
- *~/secomp/dac_checks.cf*
- *~/secomp/suid_checks.cf*
- *~/secomp/devel_checks.cf*
- */etc/.secomp_paths.*

where the tilde represents the location of the SeComp hierarchy (defaults to */etc*).

The *secomp.cf* file needs no modifications. Its use is primarily tuned to the internal requirements of the SeComp tool. The *.secomp_paths* file is created by SeComp. Do not modify this file.

5.2.1 The *~/secomp/secomp.cf* File

The *~/secomp/secomp.cf* file is the primary SeComp configuration file and contains most of the variable setting and exporting required for SeComp. This file contains a number of variable settings that describe the environment to the SeComp tool. The variables that are contained in *secomp.cf* are for internal use and should not be modified.

5.2.2 The */etc/.secomp_paths* File

The */etc/.secomp_paths* file is automatically re-created at each execution of the SeComp tool. This file will contain path information used by the internal SeComp software. Do not modify this file. If problem exists where it is unknown where the SeComp tool is located or where its report files are being deposited, a quick glance at this file will provide that information. This file is created under the */etc* directory of the executing host file system.

5.2.3 The *~/secomp/dac_checks.cf* File

This file will contain the pathnames that are to be checked in terms of the DAC permissions for its entire hierarchy. For example, adding the entry `/etc/default/*` will cause the pathname to be broken into three components `/etc`, `/etc/default`, and `/etc/default/*`. All three components will be checked for their DAC attributes and specifically check if any files in the hierarchy have group or world access permissions.

5.2.4 The *~/secomp/suid_checks.cf* File

This file will contain the pathnames that are to be checked for the possibility of the setuid bits assigned. For example, the *tftp* program should not have suid bits assigned, so adding the *tftp* pathname to this file will cause this program to be checked in this manner.

5.2.5 The *~/secomp/devel_checks.cf* File

This file will contain the development tool pathnames that are to be checked for the possibility of existence on the work station. Development tools should not be located on an operational work station.

5.2.6 The *~/secomp/runtime.cf* File

The *~/secomp/runtime.cf* file contains the SeComp runtime assignments to "hard" configure the reports and masters locations. Additionally, the *TASKS* variable is assigned in this file. The variables defined in this file will tell the SeComp program where the locations of the SeComp reports and master directories are shown next with their default settings:

- `SECOMP_REPORTDIR=${SECOMPDIR}reports` - defines the location of the SeComp *reports* directory

- `SECOMP_MASTERSDIR=${SECOMPDIR}/reports` - defines the location of the SeComp *masters* directory.

The *SECOMPDIR* variable is assigned in the *run_secomp* program file. This variable always takes the value returned from the UNIX *pwd* command, thus allowing the SeComp programs to execute any where on the file system. The preferred location is the */etc* directory.

The client may specify where the SeComp reports and masters directories may be located. If the client does not specify this, the SeComp will default to the */etc* locations that are configured by default. If the client were to specify the location of the masters directory, the *runtime.cf* file could be modified to cause this:

```
SECOMP_REPORTDIR=${SECOMPDIR}/reports
SECOMP_MASTERSDIR=/usr/masters
```

Changing *SECOMP_MASTERSDIR* in this manner will cause a master file to be referenced in the */usr/masters* directory on the client, unless, of course, the */usr* directory is actually mounted from another system.

A note about the SeComp runtime location. SeComp will read the current working directory as its runtime location and export that location to all the subprograms. For example, if the *run_secomp* program is located in the */etc/secomp* directory, the */etc/secomp* pathname becomes the runtime directory assigned to the *SECOMPDIR* variable.

In the client/server configuration, all clients will automount the same */etc/secomp* directories. This implies that if one client changes the *runtime.cf* file, that change will be reflected in all automounted views of the */etc/secomp* directories subsequent to the change.

5.3 SeComp Server Configuration for Automount

The SeComp configurations depend on the correct automount configuration on the clients and correct exporting of the SeComp hierarchy on the server for the clients.

The server's NFS file share control file is configured to export the SeComp hierarchy. The following entry will be entered as a part of the default configuration (the example assumes a Solaris platform configuration and very basic option structure):

```
share -F nfs /etc/secomp
```

5.4 SeComp Client Configuration

Each client that executes SeComp in the client-server environment must be configured to understand its environment. Each client is at least configured with the automount information that identifies the location of the SeComp server. The remaining configuration items such as the runtime, reports, and masters directories will be configured with the defaults shown in the SeComp Server Configuration section. These may be used as they are, or as discussed previously, may be configured.

5.5 SeComp Client Configuration for Automount

The SeComp client will be configured to know where the SeComp runtime directories are by modifying the */etc/auto_master* file, shown next with the default settings (the examples assume a Solaris platform):

```
/-    /etc/auto_direct
```

This points to the */etc/auto_direct* file which contains the server location information, shown next with the default settings:

```
/etc/secomp
[server_name]:/etc/secomp
```

where *server_name* will be replaced with the correct SeComp server name.

6. Administrative Issues

The SeComp tool requires some routine maintenance to avoid wasting root file system space. The two directories that require attention are:

```
/etc/secomp/reports/[hostname]
/etc/secomp/tmp
```

The reports directory will contain a *time-stamp* directory with four *SECOMP_PHASE* named sub-directories for each segment tested. It is beyond the scope of this document to determine how long to retain the report directories generated, however, some guidance is offered:

- The reports should be maintained until all segment issues have been satisfactorily resolved by the analysts and risk analysis teams.
- The reports should at some time be place into a long-term archive (tape) and stored to be accessible in case questions or clarifications come up or are required concerning the disposition of a tested segment.

Again, the issue is to come up with a standard policy for clearing the host report hierarchies.

The */etc/secomp/tmp* file contains temporary files created during the execution of the SeComp tool. The contents of this directory should be purged regularly, but after the execution of the SeComp tool is complete. As SeComp development matures, there will be fewer temporary files left once the SeComp tool execution is complete.

Appendix A. References

- *Defense Information Systems Agency (DISA), Security Checklists for the Defense Information Infrastructure (DII) Common Operating Environment (COE)*, November 1996.
- *Defense Information Infrastructure (DII) Common Operating Environment (COE) Security Compliance (SeComp) Tool System Test Descriptions (STD), Version 1.0.0.3*, May 21, 1997.
- *Defense Information Infrastructure (DII) Common Operating Environment (COE) Security Compliance (SeComp) Tool System Administrators Manual (SAM), Version 1.0.0.3*, May 21, 1997.
- *Defense Information Infrastructure (DII) Common Operating Environment (COE) Security Compliance (SeComp) Tool Software Requirements Specification (SRS), Version 1.0.0.3*, May 21, 1997.
- *Defense Information Infrastructure (DII) Common Operating Environment (COE) Security Compliance (SeComp) Tool Version Description Document (VDD), Version 1.0.0.3*, May 21, 1997.